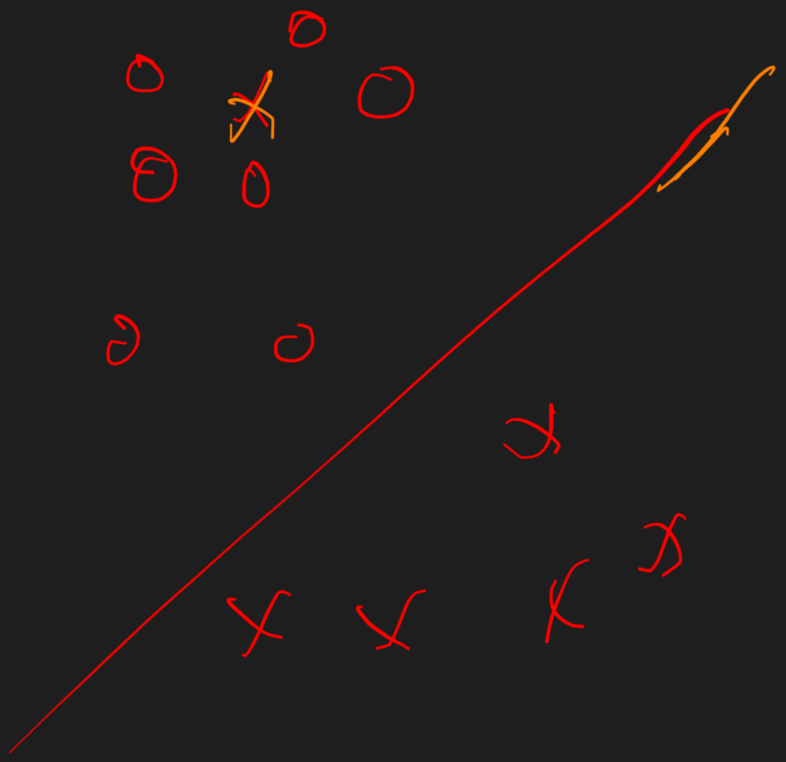
INTRODUCTION TO MACHINE LEARNING COMPSCI 4ML3

Lecture 14

Hassan Ashtiani

LINEAR CLASSIFIER

- $\hat{y}_w(x) = sign(W^T x) = 1_{W^T x \geq 0}$
- HOW TO FIND W GIVEN DATA SET Z?
  - $\widehat{W} = arg \min_{w} L_Z^{0-1}(\hat{y}_w)$
  - HOW TO MINIMIZE?



HARDNESS OF LINEAR CLASSIFICATION

- In general, finding the linear separator with minimum classification error is NP-hard (with respect to d)
- Unless….
  - Data is linearly separable! ✓
  - ..or optimize a surrogate loss function instead! ✓
- Can't we just go into higher dimensions to make the data linearly separable?

$$\ell^{0-1}(y, \hat{y}) = \begin{cases} 1 & \text{o.w} \\ 0 & \hat{y} = y \end{cases}$$

SURROGATE LOSS FUNCTIONS

- So far, we assumed the classifier returns a discrete value
  - E.g., $\hat{y}_w = sign(W^T x) \in \{0,1\}$

$sgn(w^T x)$

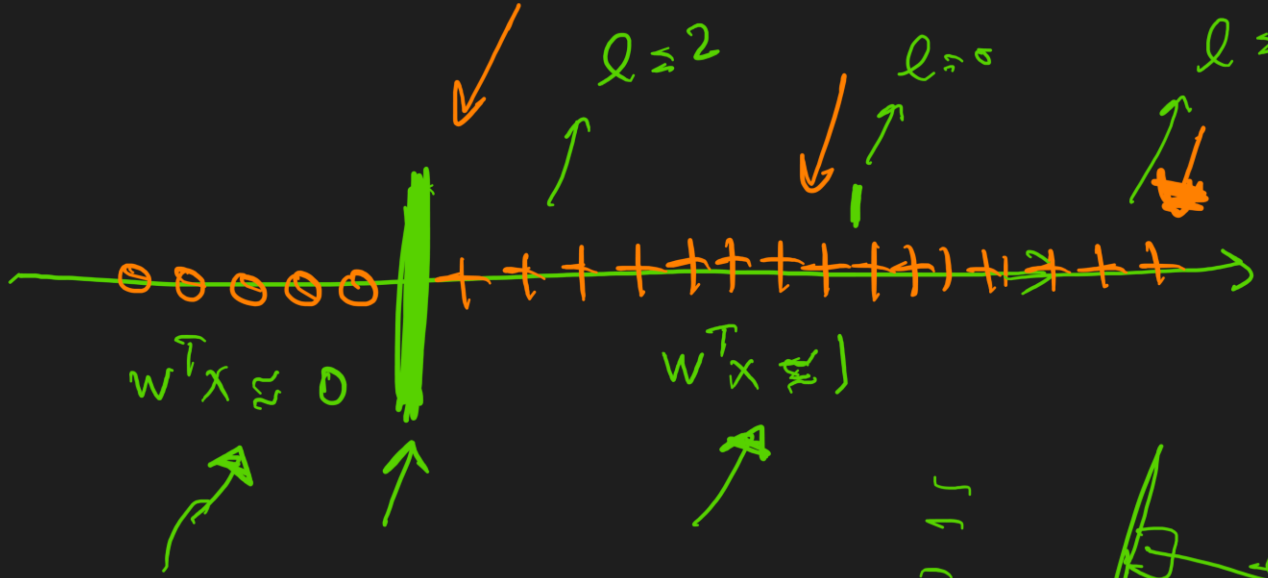
- What if the classifier's output is continuous
  - E.g., $\hat{y}_w = W^T x$
  - $\hat{y}$ will capture the "confidence" of the classifier too
- Other (continuous) loss functions
  - Margin loss, cross-entropy/negative-log-likelihood loss, …
- Potential benefits?
  - Easier to optimize
  - Mitigate Overfitting

SQUARED LOSS FOR CLASSIFICATION?

$x_1 = 10,\ y_1 = 1$

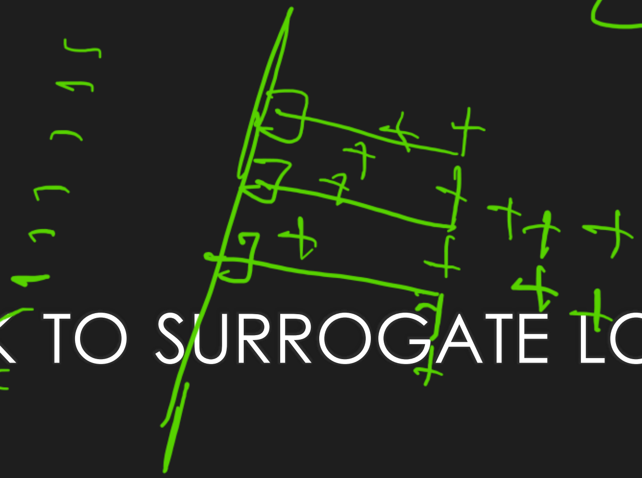
$x_2 = 11,\ y_2 = 1$

$x_3 = 8,\ y_3 = 0$



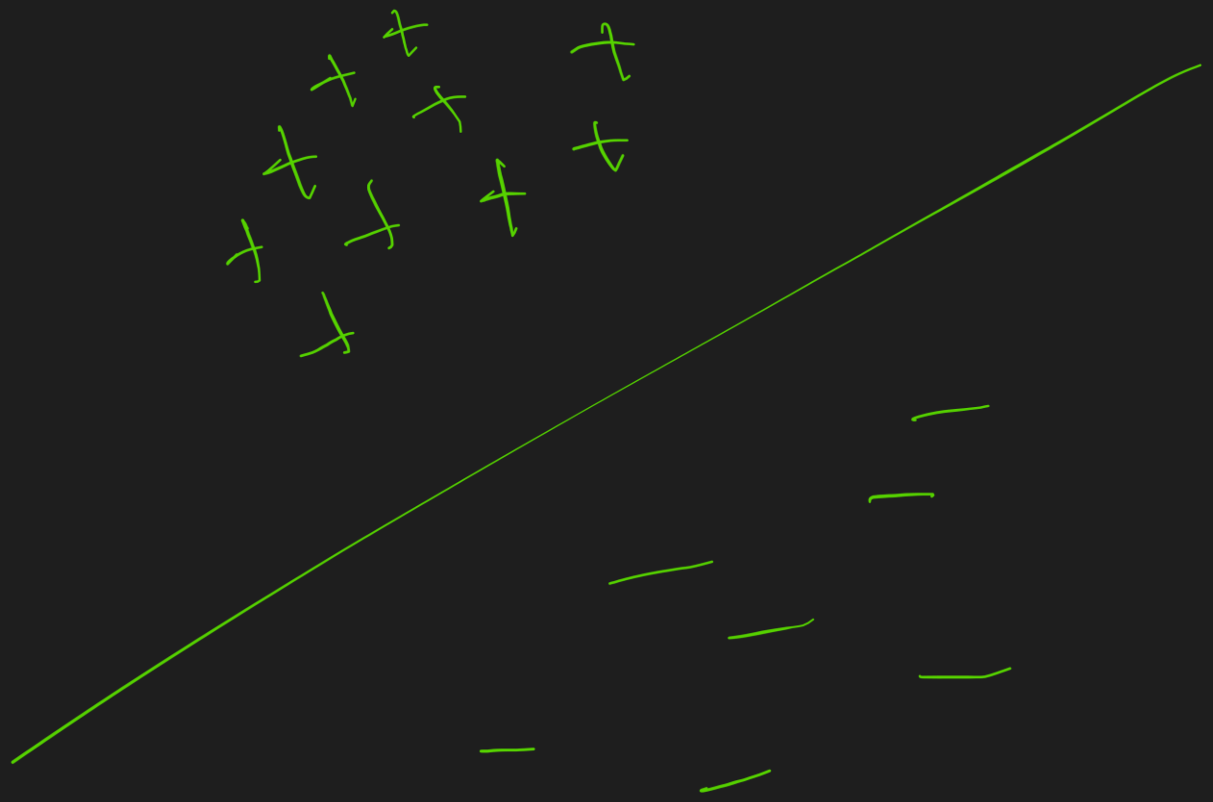
$$\sum (\hat{y}^i - y^i)^2$$
$$= \sum (w^T x^i - y^i)^2$$

$w^T x = 1$



- WE WILL GET TO BACK TO SURROGATE LOSS FUNCTIONS

LINEARLY SEPARABLE DATA



$y^i \in \{+1, -1\}$

LINEARLY SEPARABLE DATA

- A BINARY CLASSIFICATION DATA SET $Z = \{(x^i, y^i)\}_{i=1}^{n}$ IS LINEARLY SEPARABLE IF
  - THERE EXISTS W^* SUCH THAT
    - FOR EVERY $i \in [n]$ WE HAVE $\text{SGN}(< x^i, W^* >) = y^i$
    - OR EQUIVALENTLY, FOR EVERY $i \in [n]$ WE HAVE $({W^*}^T x^i) y^i > 0$
- IN OTHER WORDS, THE CLASSIFICATION ERROR ON Z IS 0
- CAN WE FIND W^* EFFICIENTLY FOR LINEARLY SEPARABLE DATA?

LINEAR PROGRAMMING

- STANDARD LP PROBLEM:

$$\max_{w \in \mathbb{R}^d} < u, w > = \sum_{i=1}^{d} u_i w_i$$

$$s.t. \quad Aw \geq v \rightarrow K \text{ linear constraints}$$

$\rightarrow u, w \in \mathbb{R}^d$

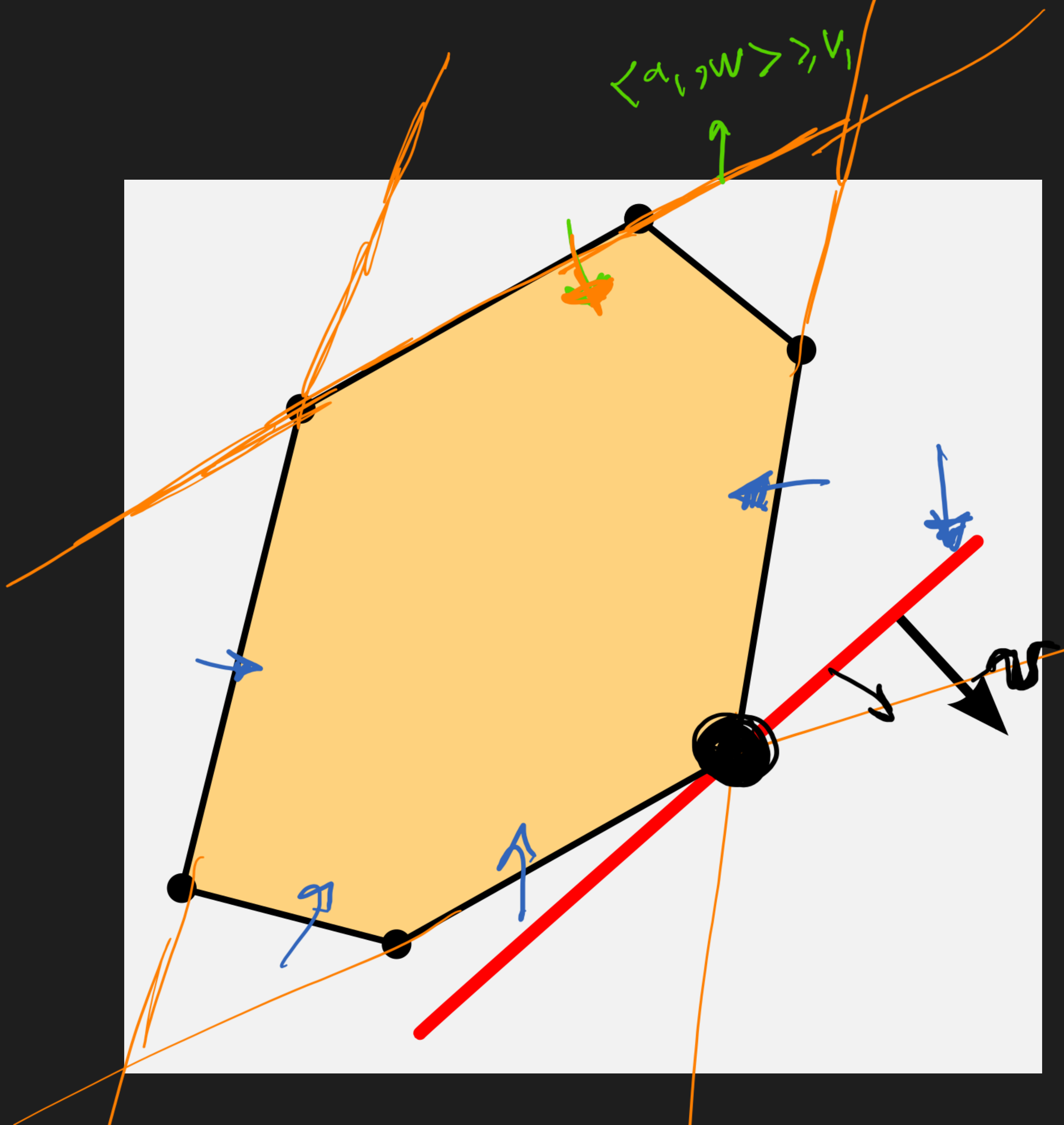
$A_{K \times d}$ matrix, $v \in \mathbb{R}^K$

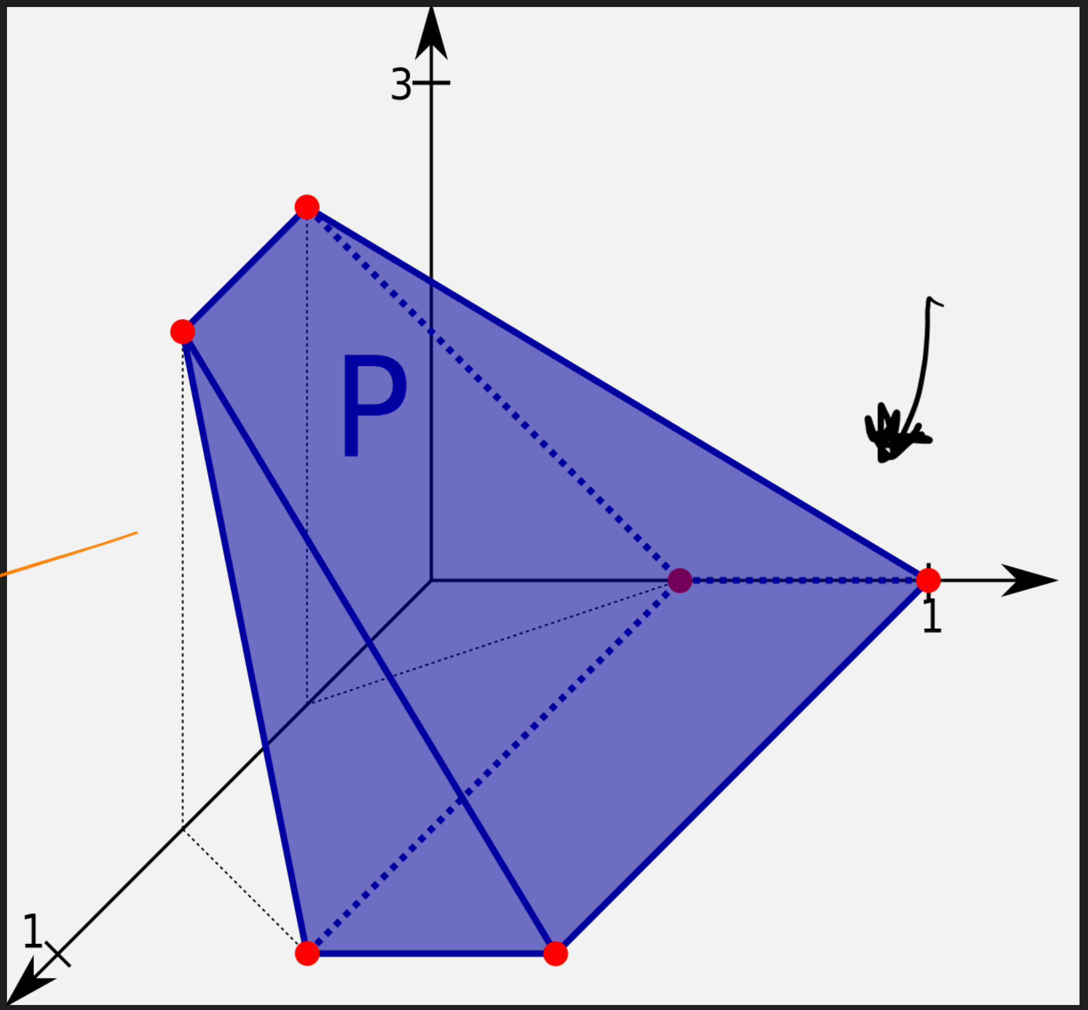
$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_K^T \end{bmatrix}_{K \times d}$$

$$\begin{bmatrix} \langle a_1, w \rangle \geq v_1 \\ \vdots \\ \langle a_K, w \rangle \geq v_K \end{bmatrix}$$

A, u, v are given.

- LP PROBLEMS CAN BE SOLVED EFFICIENTLY!


⟨a₁, w⟩ ≥ v₁


P
3
1
1

LP FOR CLASSIFICATION

$(W^{*T}x^i)y^i = 0.01$

γ

- Data is linearly separable so
  - $\exists W^*$ s.t. $\forall i \in [n]$, $(W^{*T}x^i)y^i \boxed{> 0}$
- So,
  - $\exists W^*, \gamma > 0$ s.t. $\forall i \in [n]$, $(W^{*T}x^i)y^i \boxed{\geq \gamma}$

1000

- So,
  - $\boxed{\exists W^*}$, s.t. $\forall i \in [n]$, $\boxed{(W^{*T}x^i)y^i \geq 1}$

$W^*_{new} = W^*_{old} \times \frac{1}{\gamma}$

LP FOR LINEAR CLASSIFICATION

- Define $\boxed{A = [x_j^i y^i]_{n\times d}}$
- Then finding the optimal W is equivalent to

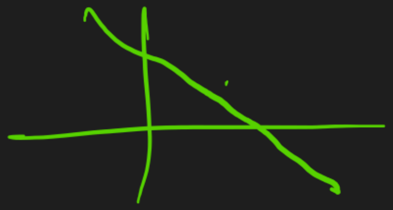
$$\max_{w\in\mathbb{R}^d} < \vec{0}, w >$$

$$s.t.\ \ Aw \geq \vec{1}$$

- We can use off-the-shelf LP solvers.
- What if the best W does not go through the origin? (It has a bias or intercept)?

$w^T x + b$

$\boxed{b=0}$



APPROACH 2: PERCEPTRON

- Proposed in 50's by Rosenblatt
- Predecessor of Neural Networks
  - Multi-layer Perceptron!

ROSENBLATT'S PERCEPTRON

Batch Perceptron

input: A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
initialize: $\mathbf{w}^{(1)} = (0, \ldots, 0)$
for $t = 1, 2, \ldots$
  if $(\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ **then**
    $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \boxed{y_i \mathbf{x}_i}$
  else
    output $\mathbf{w}^{(t)}$

- In each update, W becomes "more correct" on x^i
- https://phiresky.github.io/kogsys-demos/neural-network-demo/?preset=Rosenblatt+Perceptron

$y^i \in \{+1, -1\}$

THE GREEDY UPDATE

- IN EACH UPDATE, W BECOMES "MORE CORRECT" ON x^i:

$$\underbrace{w_{new}^T x^i y^i} = \langle \underbrace{w_{old} + y^i x^i}, x^i \rangle y^i$$

$$= w_{old}^T x^i y^i + \|x^i\|_2^2 y^i y^i$$

$$= \underbrace{w_{old}^T x^i y^i} + \underbrace{\|x\|_2^2} \gg \underbrace{w_{old}^T x^i y^i}$$

- WHAT ABOUT OTHER x^j'S?